CDS

TECHNICAL MEMORANDUM NO. CIT-CDS 96-021
December, 1996

"Constrained Nonlinear Optimal Control: A Converse HJB Approach"

Vesna Nevistic and James A. Primbs

Control and Dynamical Systems
California Institute of Technology
Pasadena, California 91125

Technical Memorandum No. CIT–CDS 96-021

Constrained Nonlinear Optimal Control: A Converse HJB Approach

Vesna Nevistić* and James A. Primbs†

December, 1996

Abstract

Extending the concept of solving the Hamilton-Jacobi-Bellman (HJB) optimization equation backwards [2], the so called converse constrained optimal control problem is introduced, and used to create various classes of nonlinear systems for which the optimal controller subject to constraints is known. In this way a systematic method for the testing, validation and comparison of different control techniques with the optimal is established. Because it naturally and explicitly handles constraints, particularly control input saturation, model predictive control (MPC) is a potentially powerful approach for nonlinear control design. However, nonconvexity of the nonlinear programs (NLP) involved in the MPC optimization makes the solution problematic. In order to explore properties of MPC–based constrained control schemes, and to point out the potential issues in implementing MPC, challenging benchmark examples are generated and analyzed. Properties of MPC-based constrained techniques are then evaluated and implementation issues are explored by applying both nonlinear MPC and MPC with feedback linearization.

1 Introduction

Determination of the optimal feedback law for nonlinear optimal control problems leads to the Hamilton–Jacobi–Bellman (HJB) partial differential equations. These equations can be solved numerically for very low state dimensions, but general methods do not exist which overcome the severe exponential growth in computation as a function of the state dimension. Thus the HJB p.d.e. itself cannot be viewed as a general practical method for nonlinear control design, and various alternative methods have been developed. These methods generally give up the optimality of the HJB solution in favor of reduced computational complexity. However, since the true optimal is not computable, there exists no systematic methodology for testing and evaluation of those approaches, and no benchmark examples which allow for fair comparison between them.

To provide a better understanding of nonlinear control design and particularly the relationship between some of the more popular nonlinear control design methodologies, a specific procedure for generating various classes of nonlinear systems for which the optimal controller is known, based on the converse HJB (CoHJB) approach, was proposed [2]. Since the CoHJB examples are constructed independently from the particular nonlinear design method, they provide insights into the original optimal control problem and can be used as benchmarks for the systematic evaluation and performance comparison of different techniques. Some test cases which illustrate the ways in which one can analyze a given design methodology are presented in [2].

*Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH), CH-8092 Zürich, Switzerland, e-mail: vesna@aut.ee.ethz.ch

†Control and Dynamical Systems, California Institute of Technology, Pasadena, California 91125, e-mail: jprimbs@cds.caltech.edu. Supported by NSF.

Because it avoids solving the HJB equation, and, in addition, naturally and explicitly handles constraints, model predictive control (MPC) is a potentially powerful alternative approach for nonlinear control design. MPC replaces the exact global HJB formulation of optimality with a necessary condition using the Euler-Lagrange formulation. While this approach is best-known for its use in off-line open loop trajectory optimization, MPC uses a receding horizon to compute an on-line feedback law. While the Euler-Lagrange formulation overcomes the exponential explosion of the HJB method, MPC requires substantial on-line computation, which has tended to limit its application to control systems with very long sample periods and slow dynamics. MPC, of course, also has the difficulty associated with all the alternatives to the HJB approach in not guaranteeing a global optimum. A particular difficulty of MPC is that its on-line algorithmic nature makes systematic analysis and evaluation of its performance problematic. While MPC has been applied widely, there are few studies comparing its performance with alternative schemes.

In this report, the CoHJB method is extended to the constrained case and used to supply benchmark examples of systems with constraints, where the optimal constrained control action is known. Referred to as the *("converse constrained HJB method" (CoCHJB))*, the underlying principle is the same as that behind the CoHJB: starting with an optimal value function, and for a given performance objective, the HJB equation is used to characterize the class of dynamics for which the assumed value function corresponds to the solution of the optimal control problem subject to the imposed constraints. Systems constructed in this way not only provide appropriate benchmarks for testing MPC, as well as other design techniques, but additionally help to further our understanding of (non)linear constrained control design. In particular, the importance of explicitly handling constraints in optimal control design is explored, demonstrating the efficiency of using constrained MPC-based control schemes, as well as their relationship with other popular nonlinear design methods.

2 Optimal and Converse Optimal Control

Consider the nonlinear system of the form[1]:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \quad f(0) = 0 \tag{1}$$

with $x \in \Re^n$, $u \in \Re^m$, and performance objective:

$$V(u) = \int_0^\infty (q(x(t)) + u^T(t)u(t))\, dt. \tag{2}$$

Definition 1 ***Optimal Control Problem:*** *Find a state-feedback control law $u^* = \Phi(x)$ such that the performance objective (2) is minimized, subject to the nonlinear system dynamics (1).*

As is well known, a standard dynamic programming argument converts the optimal control problem into the Hamilton-Jacobi-Bellman (HJB) equation:

$$\frac{\partial V^*(x)}{\partial x} f(x) - \frac{1}{4}\frac{\partial V^*(x)}{\partial x} g(x) g^T(x) \frac{\partial V^{*T}(x)}{\partial x} + q(x) = 0 \tag{3}$$

where $V^*(x)$ denotes the *value function*, given by:

$$V^*(x) = \min_u \int_0^\infty (q(x(t)) + u^T(t)u(t))\, dt \tag{4}$$

Once the HJB is solved for V^*, the optimal control is given by:

$$u^* = -\frac{1}{2} g^T(x) \frac{\partial V^{*T}(x)}{\partial x} \tag{5}$$

[1]To make the exposition concise but without loss of generality, we only consider the case of no disturbances, so the only uncertainty is the initial condition. The case with disturbances included is described in [2].

The HJB equation is difficult to solve analytically, in particular there is no efficient algorithm available when the problem dimension is high. Thus reducing the optimal control problem to the HJB equation cannot be viewed as a general, practical method.

However, by introducing the *converse optimal control problem* [2], which follows the basic idea of solving the HJB equation backwards (*"converse HJB method"*), various classes of highly nonlinear systems with possibly high state dimensions, yet for which the optimal controller is known can be constructed. In contrast to the standard optimal control problem, the converse problem starts with an optimal value function, and for a given performance objective uses the HJB equation to characterize the class of dynamics for which the assumed value function corresponds to the solution of the optimal control problem.

Definition 2 ***Converse Optimal Control Problem (CoHJB):*** *Given a performance objective (2) and a value function V^*, find the class of nonlinear systems (1) such that V^* corresponds to the solution of the optimal control problem (3). The optimal control law u^* is then determined by (5).*

Solving the CoHJB is equivalent to solving the HJB "backwards" with known value function V^*. The HJB (3) then becomes an algebraic equation in f and g, and in this way nonlinear dynamics and optimal controllers may be generated. Note that essentially any nonlinear optimal control problem of the type described above can be generated with this method. Unfortunately, this is not useful for actual design when f, g, and q are given. However, since the optimal control is known, the CoHJB provides an efficient procedure to validate the performance of different nonlinear control techniques.

It is important to note that the CoHJB is not a well defined problem for every choice of V^* and q. The following definition and theorem address the *admissibility* issue.

Definition 3 *The pair (V^*, q) is said to be* ***admissible*** *for the CoHJB problem if there exists a continuous g and f, with $f(0) = 0$ such that f, g, V^* and q satisfy the HJB (3). Furthermore, we define the set $\mathcal{A}$ to be the set of all admissible pairs (V^*, q).*

Theorem 1 *$(V^*, q) \in \mathcal{A}$ if and only if q can be factored into the product:*

$$q = V_x^* h, \quad \text{where } V_x^* = \frac{\partial V^*}{\partial x}, \tag{6}$$

for $h \in \mathbf{C}$, $h(0) = 0$.

Proof: The proof of this theorem will be given in section 5.1, as a special case of the Theorem 3, which states the admissibility conditions for the CoCHJB method. ∎

In the following, the HJB equation and the CoHJB approach will be demonstrated on some representative system descriptions [2].

2.1 Linear Systems

$$\dot{x} = Ax + Bu$$

If $q(x) = x^T Q x$ and $V^*(x) = x^T x$, the HJB equation reduces to the Riccati equation

$$A + A^T - BB^T + Q = 0,$$

which yields,

$$A = \frac{1}{2}(-Q + BB^T) + S$$

where S is any skew matrix, $(S + S^T = 0)$. The optimal control is $u^* = -B^T x$.

[2] To avoid confusion with the *inverse optimal control problem*, usually concerned with finding for what cost function some given controller for a given system is optimal, here we use the term *converse problem*.

2.2 1-D Systems

$$\dot{x} = f(x) + g(x)u \tag{7}$$

If $q(x) = x^2$ and $V^*(x) = x^2$, the HJB equation is

$$V_x^* f(x) - \frac{1}{4}(V_x^*)^2 g^2(x) + x^2 = 0.$$

Then,

$$\begin{aligned} f(x) &= -\frac{1}{2}x(1 - g^2(x)), \\ u^*(x) &= -g(x)x \end{aligned}$$

and the closed loop system is

$$\dot{x} = -\frac{1}{2}x(1 + g^2(x))$$

where $g(x)$ can be any function.

Next, consider a slightly different *converse* problem. Let $g(x) > 0$ for all $x \in \Re$, and suppose we require $u^*(x) = -x$ to be the optimal solution. The question is for what nonlinear systems is this *linear* controller optimal. Since the optimal state feedback is $u^*(x) = -\frac{1}{2}g(x)V_x^*$, then

$$\frac{\partial V^*(x)}{\partial x} = \frac{2x}{g(x)}.$$

If we take $q(x) = qx^2$ with $q > 0$, then the HJB equation is

$$\frac{2x}{g(x)}f(x) = \frac{x^2}{g^2(x)}g^2(x) - qx^2.$$

This can be solved for $f(x) = \frac{1}{2}(1-q)g(x)x$, and the closed loop system is given by $\dot{x} = -\frac{1}{2}(1+q)g(x)x$.

2.3 Nonlinear 2-D Oscillators

Consider the general form of 2-D oscillator:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \hat{f}(x) + \hat{g}(x)u, \end{cases} \tag{8}$$

with cost,

$$V(u) = \int_0^\infty (q + u^2)dt. \tag{9}$$

For this system, the HJB equation can be written as,

$$V_1^* x_2 + V_2^* \hat{f}(x) - \frac{1}{4}(V_2^* \hat{g}(x))^2 + q(x) = 0 \tag{10}$$

where $V_i^* = \frac{\partial V^*(x)}{\partial x_i}$ has been introduced for notational convenience.

Suppose that $q(x) = x_2^2$ and $V^*(x) = x_1^2 + x_2^2$, yields:

$$2x_1x_2 + 2x_2\hat{f}(x) = \frac{1}{4}(2x_2)^2\hat{g}^2(x) - x_2^2.$$

Therefore

$$\hat{f}(x) = -x_1 - \frac{1}{2}x_2(1 - \hat{g}^2(x)).$$

The dynamics are

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_1 - \frac{1}{2}x_2(1 - \hat{g}^2(x)) + \hat{g}(x)u, \end{cases}$$

and the optimal control, according to (5), is $u^*(x) = -\hat{g}x_2$.

An interesting example is $\hat{g}(x) = x_1$ which yields a Van der Pol oscillator with a stable but linearly uncontrollable equilibrium at the origin and an unstable limit cycle. Other interesting nonlinear oscillators can be constructed with other choices of $\hat{g}$ and nonlinear terms can be added to the "frequency" and so on [2].

For the 2-D oscillator, a priori constraints on the nonlinear dynamics by insisting that $\dot{x}_1 = x_2$ are imposed. Thus, in order to assure the existence of the solution to the converse problem, V^* and q cannot be chosen arbitrarily. According to Definition 3 and the discussion about admissibility given previously, the admissible set $\mathcal{A}$ for the 2-D oscillator can be explicitly characterized by the following theorem.

Theorem 2 *There exists a solution to the CoHJB problem for the 2-D nonlinear oscillator (8) for $(V^*, q) \in \mathcal{A}$, where $\mathcal{A}$ describes the admissible sets of V^*'s and q's such that $(V^*, q) \in \mathcal{A}$ if and only if:*

$$\frac{(V_1^* x_2 + q(x))}{V_2^*} < \infty, \quad \forall x \tag{11}$$

and

$$\lim_{x \to 0} \frac{(V_1^* x_2 + q(x))}{V_2^*} = 0 \tag{12}$$

If it is desired that $\hat{f} \in \mathcal{C},\ \hat{f}(0) = 0$, solving the corresponding HJB equation (10) in terms of $\hat{f}$ yields:

$$\hat{f} = \frac{1}{4}V_2^* \hat{g}^2 - \frac{(V_1^* x_2 + q(x))}{V_2^*} \tag{13}$$

From equation (13), it is clear that for continuous $\hat{g}$, and $V^* \in \mathcal{C}^1$, we need:

$$\hat{f}(0) = -\lim_{x \to 0} \frac{(V_1^* x_2 + q(x))}{V_2^*} = 0$$

Furthermore, for $\hat{f}$ to be continuous, we would impose:

$$\frac{(V_1^* x_2 + q(x))}{V_2^*} < \infty, \quad \forall x$$

Hence, for the 2-D oscillator, it only makes sense to consider V^* and q that satisfy the conditions (11) and (12).

2.4 General Nonlinear Systems

Consider the following system

$$\dot{x} = f(x) + g(x)u,$$

with $x \in \Re^n$. Suppose $q(x) = x^T Q x$ and $V^*(x) = x^T P x$ with $P > 0$, so that

$$\frac{\partial V^*(x)}{\partial x} = 2x^T P.$$

The HJB equation is

$$2x^T P f(x) - x^T P g(x) g^T(x) P x + x^T Q x = 0. \tag{14}$$

If

$$f(x) = \frac{1}{2}(g(x)g^T(x)P - P^{-1}Q)x + P^{-1}\gamma(x)$$

with $\gamma : \Re^n \to \Re^n$, $\gamma(0) = 0$ satisfying $x^T\gamma(x) = 0$ for all $x \in \Re^n$, then (14) is satisfied. In this case, the optimal control is $u^*(x) = -g^T(x)Px$. We can choose $g(x)$ and $\gamma(x)$ to make the dynamics highly nonlinear, yet the optimal V^* is quadratic and the controller is known.

3 Optimality of Feedback Linearization

An additional usage of the CoHJB approach is that it allows for the characterization of nonlinear dynamics for which a specified nonlinear control technique is optimal. Furthermore, this characterization can be used to generate systems for which a certain technique is optimal.

Here we demonstrate the CoHJB approach by exploring the optimality of feedback linearization for 1-D systems and the nonlinear 2-D oscillator. For a complete discussion on optimality of nonlinear design techniques and general results involving feedback linearization as well as Jacobian linearization and other nonlinear design techniques, see [5].

3.1 1-D Systems

Consider the 1-D type of system introduced in section 2.2 and described by (7), with $q(x) = x^2$. It can be easily shown [2] that a FL-based controller

$$u_{fl} = -\frac{1}{g(x)}[f(x) + kx]$$

is optimal if and only if $f(x)$ and $g(x)$ satisfy the following relation:

$$f^2(x) - x^2[f'(0)]^2 + x^2[g^2(x) - g^2(0)] = 0. \tag{15}$$

3.2 2-D Oscillators

Consider the general description of 2-D nonlinear oscillator introduced in section 2.3 and given by (8). Let the cost be given by (9). A FL controller for the 2-D oscillator takes the form:

$$u_{fl} = -\frac{1}{\hat{g}(x)}(\hat{f}(x) + k_1x_1 + k_2x_2) \tag{16}$$

For this controller to be optimal, u_{fl} has to be equal to u^*, or equivalently

$$-\frac{1}{\hat{g}(x)}(\hat{f}(x) + k_1x_1 + k_2x_2) = -\frac{1}{2}\hat{g}(x)V_2^* \tag{17}$$

which gives,

$$\hat{g}^2(x) = \frac{2}{V_2^*}(\hat{f}(x) + k_1x_1 + k_2x_2) \tag{18}$$

Substituting (18) in the corresponding HJB equation (10) yields,

$$V_1^*x_2 + V_2^*\hat{f}(x) - \frac{1}{4}V_2^*(2(\hat{f}(x) + k_1x_1 + k_2x_2)) + q(x) = 0 \tag{19}$$

and according to (19), $\hat{f}(x)$ is given by,

$$\hat{f}(x) = k_1x_1 + k_2x_2 - \frac{2}{V_2^*}(V_1^*x_2 + q(x)) \tag{20}$$

To find $\hat{g}(x)$, equation (20) is used to substitute for $\hat{f}(x)$ in (18), which results in,

$$\hat{g}^2(x) = \frac{4}{V_2^*}(k_1x_1 + k_2x_2) - \frac{4}{V_2^{*2}}(V_1^*x_2 + q(x)). \tag{21}$$

Additionally, for the system to be globally feedback linearizable, it is required that $\hat{g}^2(x) > 0$.

Furthermore, the FL controller u_{fl} must be designed so that the closed loop system is optimal in a neighborhood of the origin, with respect to the linearized system. This means that k_1 and k_2 must produce the same closed loop as that of the LQR solution for the linearized system. This can be achieved by requiring that k_1 and k_2 solve an appropriate Riccati equation. For details see [5].

Based on the previous discussion, the equations for generating systems for which FL is optimal can be summarized as follows:

$$\hat{f}(x) = k_1x_1 + k_2x_2 - \frac{2}{V_2^*}(V_1^*x_2 + q(x)) \tag{22}$$

$$\hat{g}^2(x) = \frac{4}{V_2^*}(k_1x_1 + k_2x_2) - \frac{4}{V_2^{*2}}(V_1^*x_2 + q(x)) \tag{23}$$

Equations (22) and (23) will be used to generate Example 2 in Section 8.1.

4 Constrained Optimal Control Problem

Consider the following nonlinear system:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \ f(0) = 0 \tag{24}$$

$x \in \Re^n, \ u \in \Re^m$, subject to input and state inequality constraints:

$$c(x, u) \leq 0 \tag{25}$$

with performance objective:

$$V(u) = \int_o^\infty (q(x(t)) + u^T(t)u(t)) \ dt \tag{26}$$

where $q : \Re^n \to \Re$ is C^1.

Definition 4 ***Constrained Optimal Control Problem:*** *Find a state-feedback control law $u^* = \Phi(x)$ such that the performance objective (26) is minimized, subject to the nonlinear system dynamics (24) and inequality constraints (25).*

The constrained optimization problem (24)–(26) can be converted into an equivalent problem given by the following HJB equation:

$$0 = \min_{u \in \Re^m} \left\{ \frac{\partial V^*(x)}{\partial x}(f(x) + g(x)u) + u^Tu + q(x) \right\} \tag{27}$$
$$\text{subject to: } c(x, u) \leq 0.$$

To solve the optimization problem (27), the first step is to perform the indicated minimization which leads to a control law of the form:

$$u^* = \psi(\frac{\partial V^*}{\partial x}, x) \tag{28}$$

The second step is to substitute (28) into equation (27), which becomes:

$$\frac{\partial V^*}{\partial x}(f(x) + g(x)\psi) + \psi^T\psi + q(x) = 0 \tag{29}$$

As described in the previous section, in the absence of constraints, the optimal control law (28) is given by (5), and the corresponding HJB equation is (3).

In the constrained case, the optimal control law (28) is obtained by solving the Kuhn-Tucker conditions [1] for the optimization problem (27):

$$2u^{*T} + \frac{\partial V^*(x)}{\partial x}g(x) + \lambda^T \frac{\partial c(x,u^*)}{\partial u} = 0 \tag{30}$$

$$\lambda^T c(x,u^*) = 0 \tag{31}$$

$$\lambda \geq 0 \tag{32}$$

Here, λ denotes the Lagrange multiplier.

Once the optimal control u^* is found, substitution in (29) for ψ results in the constrained version of the HJB equation. As emphasized earlier, the HJB equation is a nonlinear partial differential equation which is very hard to solve. The presence of constraints makes the optimal control problem even more difficult.

5 Converse Constrained Optimal Control Problem

Extending the original converse approach (CoHJB) discussed in section 2 to the constrained case, the so-called *"converse constrained optimal control problem"* is introduced.

Definition 5 ***Converse Constrained Optimal Control Problem (CoCHJB):*** *Given a performance objective (26) and a value function V^*, find the class of nonlinear systems (24) such that V^* corresponds to the solution of the constrained optimal control problem (27). The optimal control law is then given by (30)-(32).*

The basic principle behind the *"converse constrained problem (CoHJB)"* [2] is to solve the HJB "backwards" to generate an array of examples which have a known *constrained* optimal controller. In this way, insight into the original optimal control problem is provided, and since the optimal constrained controller is known, a systematic method for the testing, validation and comparison of different control techniques in the presence of constraints is established.

First the conditions for a CoCHJB problem be solvable are defined and its solution is demonstrated for single input systems with input saturation constraints.

5.1 Admissibility

In order the CoCHJB problem be solvable, a pair (V^*, q) has to be chosen according to the following definition.

Definition 6 *The pair (V^*, q) is said to be* ***admissible*** *for the CoCHJB problem if there exists a continuous g and f, with $f(0) = 0$ such that f, g, V^* and q satisfy the HJB (29). Furthermore, we define the set $\mathcal{A}$ to be the set of all admissible pairs, (V^*, q).*

If we assume that q is at least $\mathcal{C}$ and V^* is at least $\mathcal{C}^1$, then we have the following theorem concerning admissibility [4].

Theorem 3 *Consider the CoCHJB with saturation constraint, $\|u\|_\infty \leq \alpha$. Then $(V^*, q) \in \mathcal{A}$ if and only if q can be factored into the product:*

$$q = V_x^* h \tag{33}$$

for $h \in \mathcal{C}$, $h(0) = 0$.

Proof: ($\Rightarrow$) Assume $(V^*, q) \in \mathcal{A}$, then there exist a g and f, $f(0) = 0$ such that the HJB eq. is satisfied:

$$V_x^*(f + gu^*) + u^{*T}u^* + q = 0 \tag{34}$$

where $u^* \in \mathcal{C}$ describes the optimal controller. Hence,

$$\begin{aligned} q &= -(V_x^*(f + gu^*) + u^{*T}u^*) &(35)\\ &= V_x^*(-(f + gu^* + \frac{V_x^{*T}u^{*T}u^*}{\|V_x^*\|^2})) &(36)\end{aligned}$$

if $-(f + gu^* + \frac{V_x^{*T}u^{*T}u^*}{\|V_x^*\|^2}) \in \mathcal{C}$, then we are done since this is our desired h. To show that this is indeed the case, consider two situations:

- Constraint Inactive: If the constraint is inactive, then $\lambda = 0$ and the optimal control is given by:

  $$u^* = -\frac{1}{2}g^T V_x^{*T}$$

  in this case (35) is:

  $$q = V_x^*(-(f - \frac{1}{4}gg^T V_x^{*T}))$$

  So $h = -(f + gu^* + \frac{V_x^{*T}u^{*T}u^*}{\|V_x^*\|^2}) = -(f - \frac{1}{4}gg^T V_x^{*T}) \in \mathcal{C}$ and using that $V_x^*(0) = 0$ and $f(0) = 0$ gives $h(0) = 0$.

- Constraint Active: In this case, it is critical to note that it is impossible to have $V_x^* = 0$ while any of the constraints are active. This is a simple consequence of analyzing (30–32) when any element of u is equal to α or $-\alpha$. Hence we must have that $V_x^* \neq 0$ or $\|V_x^*\|^2 \neq 0$. Then again we have $-(f + gu^* + \frac{V_x^{*T}u^{*T}u^*}{\|V_x^*\|^2}) \in \mathcal{C}$.

Therefore, we have established that $h = -(f + gu^* + \frac{V_x^{*T}u^{*T}u^*}{\|V_x^*\|^2})$ satisfies the required properties.

($\Leftarrow$) Assume $q = V_x^* h$, with $h \in \mathcal{C}$, $h(0) = 0$. For any $g \in \mathcal{C}$, let,

$$f = -(h + gu^* + \frac{V_x^{*T}u^{*T}u^*}{\|V_x^*\|^2}) \tag{37}$$

hence $f \in \mathcal{C}$, $f(0) = 0$ (Since $u^* = -\frac{1}{2}g^T V_x^{*T}$ in a neighborhood of $u^* = 0$) and f, g, V and q satisfy the HJB. Therefore $(V^*, q) \in \mathcal{A}$. ∎

The above result can easily be extended to more general forms of constraints.

Theorem 4 *Consider the CoCHJB with constraint $c(x,u) \leq 0$ such that $V_x^* = 0$ implies that $c(x,u) < 0$, (i.e., the constraint is* inactive *whenever $V_x^* = 0$). Furthermore, assume that $c(x,u)$ is such that u^* resulting from the solution of (30)–(32) is continuous. Then $(V^*, q) \in \mathcal{A}$ if and only if q can be factored into the product $q = V_x^* h$ with $h \in \mathcal{C}$, $h(0) = 0$.*

5.2 Single Input Systems With Saturation Constraints

Consider a nonlinear system (24) with scalar input u, and saturation constraint:

$$|u| < \alpha \tag{38}$$

which is equivalent to:

$$c(x,u) = \begin{bmatrix} u-\alpha \\ -u-\alpha \end{bmatrix} \tag{39}$$

Then the constrained optimal u^*, according to (30)–(32) is characterized by:

$$2u^* + \frac{\partial V^*}{\partial x} g + \lambda_1 - \lambda_2 = 0 \tag{40}$$

$$\lambda_1(u^* - \alpha) + \lambda_2(-u^* - \alpha) = 0 \tag{41}$$

$$\lambda_1 \geq 0 \tag{42}$$

$$\lambda_2 \geq 0 \tag{43}$$

Solving the above equations for $u^* = \psi(\frac{\partial V^*}{\partial x}, x)$ leads to the following result:

Theorem 5 *The optimal controller for a single input system (24) under saturation (38) is given by:*

$$u^* = sat_\alpha(-\frac{1}{2} g^T(x) \frac{\partial V^{*T}(x)}{\partial x}), \tag{44}$$

where $sat_\alpha(\cdot)$ denotes the saturation operator:

$$sat_\alpha(z) = \begin{cases} \alpha & z > \alpha \\ z & -\alpha \leq z \leq \alpha \\ -\alpha & z < -\alpha \end{cases} \tag{45}$$

The corresponding HJB is given by:

$$\frac{\partial V^*}{\partial x}(f + gu^*) + u^{*T} u^* + q(x) = 0. \tag{46}$$

Proof:

- Constraint Inactive: $|u^*| < \alpha$

  When $\lambda_1 = \lambda_2 = 0$, equation (41) is trivially satisfied, indicating that the optimal u^* does not occur with the constraint active. Equation (40) becomes,

  $$2u + \frac{\partial V^*}{\partial x} g = 0 \tag{47}$$

  allowing u^* to be solved for explicitly as,

  $$u^* = -\frac{1}{2} g^T \frac{\partial V^*}{\partial x}^T \tag{48}$$

- Constraint Active: $u^* = \alpha$

  Here $\lambda_1 > 0$, and $\lambda_2 = 0$, which is equivalent, by equation (41), to having $u^* = \alpha$. Then equation (40) becomes:

  $$2\alpha + \frac{\partial V^*}{\partial x} g + \lambda_1 = 0 \tag{49}$$

  Since λ_1 is to be strictly positive, (49) implies,

  $$-\frac{1}{2} g^T \frac{\partial V^*}{\partial x}^T > \alpha \tag{50}$$

  which defines the region in which the control $u^* = \alpha$ is optimal.

- Constraint Active: $u^* = -\alpha$

  Here $\lambda_1 = 0$, and $\lambda_2 > 0$. Equation (41) then states that $u^* = -\alpha$ and equation (40) becomes,

$$-2\alpha + \frac{\partial V^*}{\partial x} g - \lambda_2 = 0 \tag{51}$$

  Since λ_2 is strictly positive, it implies,

$$-\frac{1}{2} g^T \frac{\partial V^{*T}}{\partial x} < -\alpha \tag{52}$$

■

A careful inspection of the arguments involved in the proof of Theorem 5 indicates that identical arguments hold in the multi-input case. This leads to the following result.

Theorem 6 *The optimal controller for a multi-input system (24) under the saturation constraint, $||u||_\infty \leq \alpha$, is given by:*

$$u^* = sat_\alpha(-\frac{1}{2} g^T(x) \frac{\partial V^{*T}(x)}{\partial x}), \tag{53}$$

The corresponding HJB is given by:

$$\frac{\partial V^*}{\partial x}(f + gu^*) + u^{*T} u^* + q(x) = 0. \tag{54}$$

Equation (54), when thought of in terms of f and g, can be used to characterize all possible solutions of the CoCHJB. This is used later to generate illustrative examples for which the constrained optimal controller (53) is known.

As a benchmark, we consider the single input 2-D oscillator subject to saturation constraints.

5.3 Benchmark: 2-D Oscillators

Consider the 2-D oscillator introduced in section 2.3 and described by (8), now subject to the saturation constraint $|u| < \alpha$. Choosing $q(x) = x_2^2$, the performance objective (9) becomes:

$$V(u) = \int_0^\infty (x_2^2 + u^2) dt. \tag{55}$$

The solution to the CoCHJB for the 2-D oscillator is derived according to the results described in Section 5.2. Assuming the known optimal value function V^*, equations (44) and (46) yield the constrained optimal control law for the 2-D oscillator:

$$u^* = sat_\alpha(-\frac{1}{2} \hat{g} V_2^*) \tag{56}$$

and corresponding dynamics:

$$\hat{f} = -\frac{(V_2^* \hat{g} + u^*) u^* + V_1^* x_2 + q}{V_2^*} \tag{57}$$

6 Constrained Nonlinear MPC Formulation

An MPC algorithm is conventionally formulated in discrete time by solving an on–line open loop finite horizon optimal control problem at each sampling time k, respecting the following objective function $\mathcal{J}$:

$$\min_{\mathcal{U}_k} \left[\sum_{i=1}^{P} q(x_{k+i}) + \sum_{i=0}^{M-1} u_{k+i}^T u_{k+i} \right] \tag{58}$$

subject to:

$$c(x,u) \leq 0 \tag{59}$$
$$\dot{x} - f(x) - g(x)u = 0 \tag{60}$$

Here:

x_{k+i}	:	predicted state vector at time $k+i$ based on the states x_k at time k, obtained by using prediction model (60)
$\mathcal{U}_k$	:	control sequence $u_{k+i},\ i = 0, ..., M-1$ computed by the optimization algorithm at time k; $u_{k+i} = 0$ for $i \geq M$; u_k is the control move to be implemented at time k.
$M,\ P$	:	*input* (*control*) and *output* (*prediction*) horizon, respectively; $M \leq P$.

Since constraints are directly included in the optimization, MPC is considered as the only methodology which deals with both input and output constraints explicitly. Although the optimization problem solved at time k results in an optimal sequence of M present and future control moves, only the first control move, u_k, is implemented on the real plant over the time $[k, k+1]$. At time step $k+1$, the horizons M and P are shifted ahead by one step, and a new optimization problem with new initial condition x_{k+1}, is solved. This kind of implementation is known as the *receding* or *moving horizon* approach.

Determining an optimal open-loop control for a given initial state is a relatively simple task, and makes the MPC algorithms attractive in many situations. By repeatedly solving an on-line open-loop optimization for the current state, and applying the minimizing control for a short time before repeating the procedure, MPC does not require the construction of diffeomorphic state-feedback transformations and avoids solving the HJB equation. Essentially, instead of being determined for each state, the value function is calculated for the sequence of states actually encountered.

Various MPC–based methods have been employed to solve the constrained control problem for nonlinear systems. Basically, three approaches exist:

1. Nonlinear MPC (NLMPC),
2. MPC in combination with feedback linearization,
3. Approximate MPC techniques (gain-scheduled linear MPC).

Since NLMPC and MPC+FL will be applied to the examples presented in this report, some of the basic issues related to these methods will be briefly discussed.

6.0.1 NLMPC Technique

The standard nonlinear MPC (NLMPC) technique and its modifications use nonlinear models for prediction according to (58)–(60), which generally result in non–convex nonlinear programs (NLP), even if the cost function and constraint sets are convex.

An essential issue, both theoretical and practical, is whether the optimization can be successfully employed in NLMPC. The important distinction in *nonlinear programs (NLP)* is not *linear/nonlinear*, but rather *convex/nonconvex*. If the resulting nonlinear optimization problem is convex (e.g. for the *linear* system, *convex* cost function and *convex* I/O constraints), there exist methods which ensure convergence to a global minimum, which is unique if the performance criterion is strictly convex. Moreover, by exploiting duality, lower and upper bounds for the optimal cost are easily computed.

However, if the system to be controlled is *nonlinear*, even if the cost function and constraint sets are convex, the control problem will be, in general, a *nonconvex* nonlinear optimization problem. Therefore, finding a global optimum is a very difficult and computationally very demanding task, if possible at all. In other words, nonconvexity makes the solution of the NLP uncertain. This issue will be discussed later on, and clearly shown on a simple example.

The extensive computational requirements of the NLMPC method are a serious obstacle for its successful industrial implementation. Also, not many advances have been made in understanding their stability and performance properties. Therefore, the basic motivation behind developing the alternative approaches referred to above was to reduce the MPC problem to a quadratic program (QP), for which efficient software exists, and to simplify the stability and performance analysis.

6.1 MPC+FL Technique

The MPC+FL technique (Fig.1) attempts to gain computational efficiency by feedback linearizing the plant and restating the MPC problem in the new linearized coordinates [3]. The use of a linearized prediction model may reduce the NLP in the NLMPC formulation to a quadratic program (QP) for the FL system, which can have a dramatic effect on efficiency. The basic difficulty with the MPC+FL method is due to the fact that the original optimization problem for the nonlinear system subject to linear constraints on the input u, given by: $-\alpha \leq u \leq \alpha$, has been transformed into an optimization problem for a linear system subject to MPC constraints on the new input v, described by: $-\alpha \leq \xi(x) + \eta(x)v \leq \alpha$, which are *state dependent* and generally *nonlinear*. This new optimization problem is not necessarily easier to solve, unless the nonlinear constraints are convex. It is also important to note that the objective function used in MPC+FL is given in terms of the new control variable v:

$$\mathcal{J}_{fl} = \min_{\mathcal{V}_k} \left[\sum_{i=1}^{P} q(x_{k+i}) + \sum_{i=0}^{M-1} v_{k+i}^T R v_{k+i} \right] \tag{61}$$

where it has been assumed that there were no changes in the state coordinates due to FL, and R is a tuning parameter.

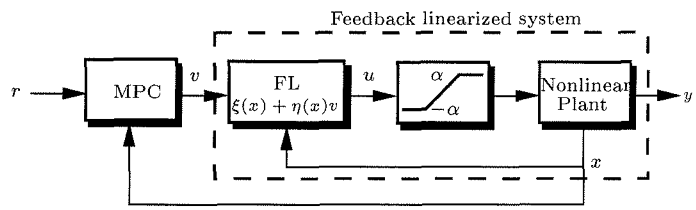

Figure 1: MPC+FL control structure

7 Implementability issues in MPC

In this section some basic problems which arise in the implementation of MPC, such as global vs. local optimum, and the influence of disturbances, are described. It is easy to construct examples which illustrate important MPC implementation issues. In particular, simple (even 1-dimensional) examples can be generated where there are local optima with arbitrarily higher cost than the global optimum. Also it is easy to construct examples where ignoring the possible effects of future disturbances in choosing the control action, as is common in MPC, leads to arbitrarily bad results. This is in contrast with the linear case, where the gap can be bounded. In what follows, we present examples demonstrating both of these issues.

7.1 Optimality/Feasibility Issue

As mentioned before, an essential issue in MPC is whether the optimization can be successfully employed. It is very easy to construct examples which clearly illustrate the problems encountered due to failures of the optimization scheme used in MPC in finding the global optimum. The following is one such example.

7.1.1 System description

Let's consider a simple discrete-time scalar system given by:

$$x_{k+1} = f(x_k) + u_k. \tag{62}$$

and subject to a state constraint at the end of the output horizon

$$x_{k+P} = 0,$$

which is the standard formulation of finite horizon MPC in order to insure nominal stability.

The cost to be minimized at each time step k is quadratic and described by:

$$J_k = \sum_{i=0}^{P} x_{k+i}^2 + \sum_{i=0}^{M-1} u_{k+i}^2 \tag{63}$$

For simplicity let's consider the system (62) for $k = 0$, taking the input and output horizon to be $M = P = 2$. Using the end constraint

$$x_2 = f(x_1) + u_1 = 0,$$

the future control move u_1 is given by:

$$u_1 = -f(x_1)$$

and current control move u_0 is:

$$u_0 = x_1 - f_0$$

where $f_0 = f(x_0)$, which is fixed.

The expression for the cost (63) can be rewritten in term of the current control move u_0:

$$J = x_0^2 + (f_0 + u_0)^2 + u_0^2 + (f(f_0 + u_0))^2, \tag{64}$$

or in term of x_1 as follows:

$$J = x_0^2 + x_1^2 + (x_1 - f_0)^2 + f(x_1)^2 = J_0 + f^2(x_1) \tag{65}$$

where $J_0 = x_0^2 + x_1^2 + (x_1 - f_0)^2$.

The purpose of expressing the cost in this way is to make obvious the choice of f such that multiple local optima appear. Since the first three terms of the cost (J_0) are quadratic and define a parabola with a minimum at $x_1 = \frac{f_0}{2}$ or equivalently $u_0 = -\frac{f_0}{2}$ independent of the function f, by a proper choice of the function f, we are able to make this optimum be only local for the overall cost J. Moreover, other local minima can be created.

An example of such an f is presented next.

7.1.2 Example

To illustrate the idea previously described, we generated the function $f(x)$ described by:

$$f^2(x) = \begin{cases} 0 & \text{for} \quad x < a \\ \frac{\alpha}{(b-a)}(x-a) & \text{for} \quad a \leq x < b \\ \alpha & \text{for} \quad b \leq x < c \\ -\frac{\alpha}{(d-c)}(x-d) & \text{for} \quad c \leq x < d \\ 0 & \text{for} \quad d \leq x < e \\ \frac{\alpha}{(x_0-e)}(x-e) & \text{for} \quad x \geq e \end{cases} \tag{66}$$

Figure 2 shows the quadratic portion of the cost, J_0, followed by the term we designed, $f^2(x)$, and finally the total cost J.

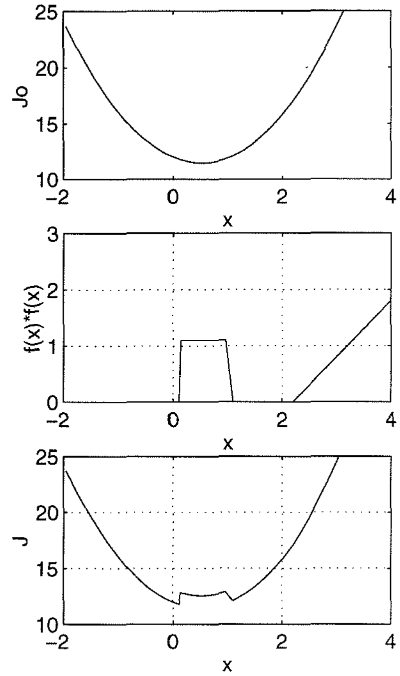

Figure 2: Top: J_0, Middle: $f^2(x)$, Bottom: Total Cost J

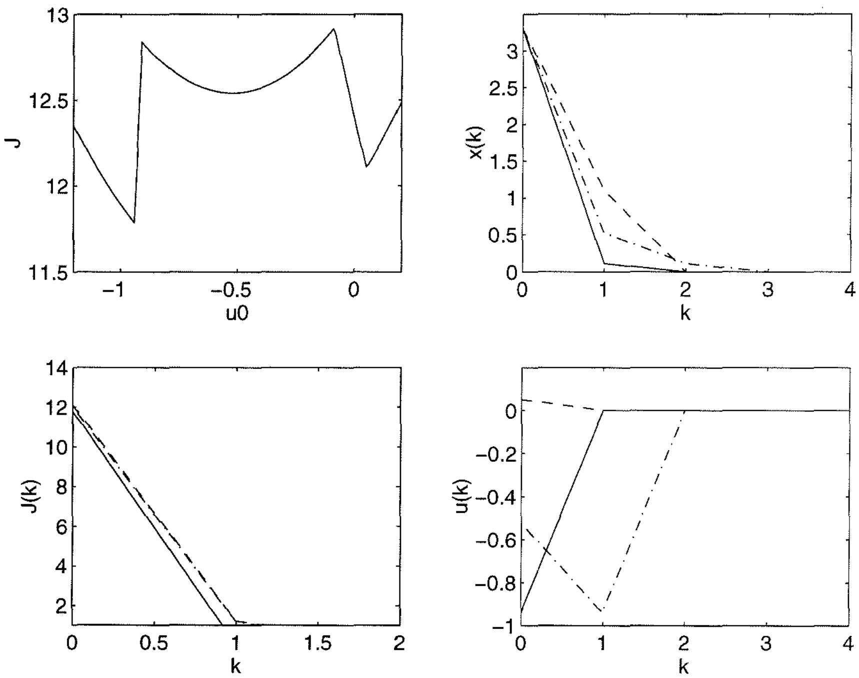

Figure 3: Top Left: Cost J, Top Right: State trajectories, Bottom Left: Progression of Cost $J(k)$, Bottom Right: Control Action $u(k)$

The parameters chosen to generate the function $f(x)$ are: $\alpha = 1.1$, $a = \frac{1}{10}\alpha$, $b = \frac{1}{8}\alpha$, $c = \frac{7}{8}\alpha$, $d = \alpha$, $e = 2\alpha$, and the initial condition $x_0 = 3\alpha$.

Applying the MPC algorithm to the system described by (66), and testing its behavior for various initial guesses for the control action u_0, it can be shown that the optimization procedure either fails by ending up in a local optimum, or finds the global optimal solution, achieving the optimal cost J. From the plots of the cost function it is obvious that due to the presence of the two local minima, optimization schemes can fail in finding the global optimum.

Figure 3 shows obtained trajectories for different starting points, as well as the computed control action and the characteristic receding horizon cost.

Remarks:

- Starting from the initial guess $u = -2$, the first optimization performed finds a global optimum $u_0 = -0.9388$, and the cost for this case is minimal as shown in fig. 3. *(solid line)*
- Starting from the initial guess $u = -0.5$, the solution found by the first optimization is $u = -0.5244$ which, according to the graphical representation $J(u_0)$ represents a local minimum *(dashdot line).*
- Starting from the initial guess $u = 0$, the first computed control action found by the optimization scheme is $u_0 = 0.0512$, which is another local optimum *(dashed line).*

It is clear that the problems in finding/missing a global optimum described above are due to the optimization technique itself, and not due to the receding horizon implementation. However, it is important to point out that similar effects and failures are possible due to short horizon lengths used in the MPC on–line optimization formulation.

7.2 Influence of disturbances

Typically, disturbances are not explicitly taken into account in the standard MPC formulation. A simple example which clearly indicates the importance of explicitly considering disturbances in controller design for nonlinear systems is shown.

7.2.1 Problem Statement

Let's consider a simple discrete-time scalar system given by:

$$x_{k+1} = f(x_k) + g_1(x_k)u_k + g_2(x_k)w_k. \tag{67}$$

where u_k and w_k describe the control action and disturbances, respectively. f,g_1 and g_2 are given by:

$$f = \begin{cases} 0 & x \leq 0 \\ -x & x > 0 \end{cases} \tag{68}$$

$$g_1 = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases} \tag{69}$$

$$g_2 = \begin{cases} 1 & x \leq 0 \\ 0 & x > 0 \end{cases} \tag{70}$$

The cost to be minimized at each time step k is quadratic and described by:

$$J_k = \sum_{i=0}^{P} x_{k+i}^2 + \sum_{i=0}^{M-1} u_{k+i}^2 \tag{71}$$

For simplicity let's consider the system (67) for $k = 0$, taking the input and output horizon to be $M = P = 2$. First, we consider an MPC formulation, without explicitly considering the disturbance, w_k.

Consider the case where $x_0 > 0$ and assume that $u_0 < x_0$. Since

$$x_1 = -x_0 + u_0 \tag{72}$$

then $x_1 < 0$, which implies that on the second step, $u_1 = 0$, (since $g_1 = 0$ for $x < 0$), and $x_2 = 0$. Hence the cost can be written out explicitly as:

$$J = x_0^2 + x_1^2 + u_0^2 = 2x_0^2 - 2x_0u_0 + 2u_0^2 \tag{73}$$

which attains a minimum for $u_0 = \frac{x_0}{2}$ with a cost of $\frac{3}{2}x_0^2$. Therefore, the MPC solution after the first iteration will be $u_0 = \frac{x_0}{2}, u_1 = 0$.

On the other hand, if $u_0 \geq x_0$ then:

$$J = x_0^2 + u_0^2 + ... \geq x_0^2 + x_0^2 > \frac{3}{2}x_0 \tag{74}$$

Hence, the MPC solution should be given by the first case with $u_0 < x_0$ which means the MPC solution will be $u_0 = x_0/2$ and $u_1 = 0$.

At the second MPC iteration, $x_1 < 0$ and the control has no effect. Furthermore, $x_2 = 0$. Hence the MPC solution will be $u_1 = 0, u_2 = 0$

Now if we consider the effects of the disturbance on our MPC solution, which is $u_0 = x_0/2, u_1 = 0$, then it is clear that since x_1 will be negative, the disturbances will completely influence the performance of the system.

If disturbances are taken into account, then it is clear that we should always keep $x_k \geq 0$, otherwise we lose control action and disturbances take over. Hence we should choose $u_0 \geq x_0$. To minimize the cost, the optimal is to take $u_0 = x_0$ and the cost (74) is then equal to $2x_0^2$. This example illustrates that not only MPC, but any technique that does not explicitly consider disturbances can perform arbitrarily bad in their presence.

8 Examples and Simulation Results

To illustrate the converse approach, various nonlinear examples with the known optimal solution, generated by CoHJB and CoCHJB, are presented in this section. Then different nonlinear control techniques are applied and their performances compared. Here we are primarily interested in the investigation of the MPC-based control techniques, i.e., how close is an MPC solution to the real optimal solution, and how MPC behaves in comparison with other less sophisticated nonlinear techniques. Both NLMPC and MPC+FL are implemented in the unconstrained and constrained case, and their properties are then compared on some representative examples. Performances are evaluated with respect to the cost achieved by each particular method[3].

8.1 Examples: Generated by CoHJB Approach

Here two examples, constructed by using the CoHJB approach, for which the *unconstrained* optimal control is known, will be presented. However, the properties of the applied control algorithms will be compared also in the presence of input constraints.

[3]Recall that the MPC controller design can be done in two different ways: (*i*) including the constraints explicitly in the optimization ("constrained MPC"), (*ii*) respecting the constraints only as a saturated input ("MPC as an LTI controller"). In the second case, if the constraints are not included into optimization, the MPC controller represents an LTI controller and computations are much simpler. For some cases where the difference between these two approaches is not significant, in order to avoid unnecessary computational burden, it is reasonable to treat the constraints just as saturation. The preferable method depends on the tuning parameters chosen, as well as on the concrete problem to be solved and the specified requirements. The name "MPC" is, however, usually associated with an explicit handling of the constraints.

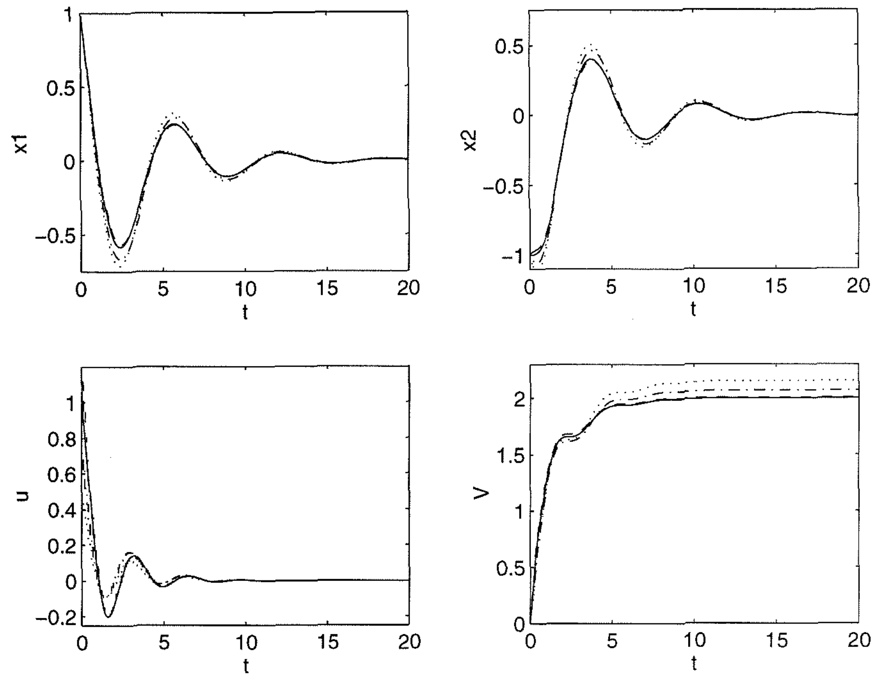

Figure 4: Unconstrained case: Optimal (solid) vs. NLMPC (P=50 : dashed, P=10 : dot-dashed, P=5 : dotted)

8.1.1 Example 1

An interesting simple example based on the oscillator description (8) is generated by taking $\hat{g}(x) = x_1$, which yields a Van der Pol oscillator with a stable but linearly uncontrollable equilibrium at the origin and an unstable limit cycle:

$$\left\{ \begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_1 - \frac{1}{2}x_2(1 - x_1^2) - x_1^2 x_2. \end{array} \right. \tag{75}$$

The value function taken was $V^* = x_1^2 + x_2^2$, and the optimal controller is $u^* = -x_1 x_2$.

The NLMPC strategy is then applied to this system[4], and the achieved performance is compared with the optimal nonlinear controller for both the unconstrained case and the case of saturating input u. Responses for initial condition $x_0 = [1 \ -1]^T$ are shown in figures 4 and 5 for the unconstrained and constrained case ($|u| \leq \alpha, \ \alpha = 0.1$), respectively.

A brief description of observations follows:

- **Unconstrained case:** It is clear that by increasing the prediction horizon, NLMPC approaches the nonlinear optimal controller. For $P = 50$, it is very difficult to notice any difference in responses and in the cost achieved.

- **Constrained case:** Even for the very tight saturation values considered here ($\alpha = 0.1$) there is no noticeable difference between performances obtained by NLMPC and the nonlinear optimal controller, as well as between responses of NLMPC obtained for different horizon lengths.

The results are summarized in Table 1. Although NLMPC deals with constraints explicitly by including them directly into the on-line optimization, for the system (75) describing the Van der Pol oscillator, this

[4]Since this system is not feedback-linearizable, the MPC+FL approach cannot be analyzed here.

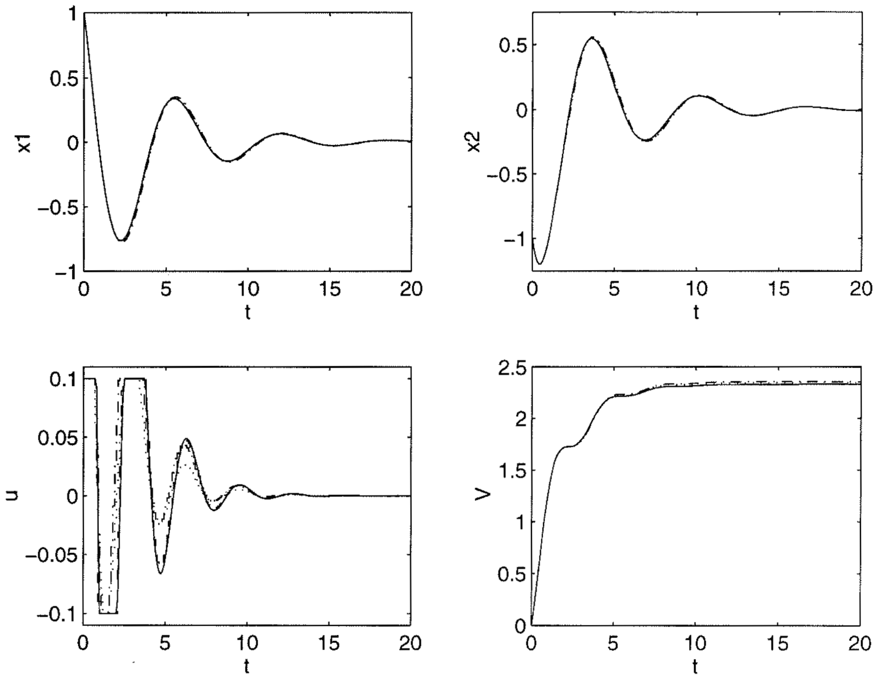

Figure 5: Constrained case: Optimal (solid) vs. NLMPC (P=50 : dashed, P=10 : dot-dashed, P=5 : dotted)

Controller	**Cost** (unsat)	**Cost** (sat)
OPTIMAL	2.0000	2.3301
NLMPC ($M = 2,\ P = 50$)	2.0063	2.3308
NLMPC ($M = 2,\ P = 10$)	2.0708	2.3576
NLMPC ($M = 2,\ P = 5$)	2.1527	2.3485

Table 1: Example 1; Results

does not result in any better performance if compared to the optimal controller, which was designed for the unconstrained case and which considers the constraints just by saturating the input. Therefore it is reasonable to expect that also the other nonlinear unconstrained techniques behave well in the presence of the constraints. Since these methods are usually computationally less demanding, using NLMPC for this type of dynamics would not be justified. However, there exist different classes of (non)linear systems, where the MPC-based techniques show superior behavior versus all other known control methods. Very often the control techniques which do not take constraints explicitly into account perform poorly or lead to instability, indicating that constrained MPC-based algorithms appear to be necessary to cope with constraints. An example which illustrates these effects follows.

Since this example is constructed using the CoHJB method, which provides the known *optimal unconstrained* controller, the true "optimal" solution under constraints is not known. Therefore a discussion how close or how far the solution found by NLMPC or any other particular approach is to the global optimum is not possible. On the other hand, as will be explored later on, the CoCHJB approach allows us exactly this kind of assessment and comparison.

8.1.2 Example 2

Using equations (22) and (23) from the results of Section 3 on optimal feedback linearization for the 2-D oscillator, the following example is generated.

Assume that $q(x) = x_2^2$ and the value function V^* is known

$$V^* = e^{x_1^2+x_2^2} - 1$$

Hence, for feedback linearization to be optimal, we use (22) and (23) with $k_1 = k_2 = 1$ to obtain $\hat{f}(x)$ and $\hat{g}(x)$:

$$\begin{aligned} \hat{f}(x) &= -x_1 + x_2 - e^{-(x_1^2+x_2^2)}x_2 \\ \hat{g}(x) &= \sqrt{2e^{-(x_1^2+x_2^2)} - e^{-2(x_1^2+x_2^2)}} \end{aligned} \tag{76}$$

In this case both the optimal, and feedback linearized controller are equal and given by:

$$u = -\frac{1}{2}\hat{f}(x)V_2^*.$$

Note that the 2-D oscillator, as defined in (8) is full state feedback linearizable.

For this example, the optimal/FL controller, NLMPC, and the MPC+FL are compared.

- **Unconstrained case:** Figure 6 shows a comparison of the optimal/FL controller, NLMPC, and MPC+FL for the initial condition [1 0]. The upper plots compare the states trajectories as a function of time, while the lower left plot shows the corresponding control action, and the lower right plot shows the progression of the cost as a function of time. As expected, NLMPC (dashed) achieves a trajectory and cost similar to that of the optimal/FL controller (solid). More surprising might be the fact that the MPC+FL approach (dotted) also performs close to the optimal. In fact, it achieves a cost below that of NLMPC, despite not using the original performance objective. In addition, it requires much less computational effort than its NLMPC counterpart.

  In hindsight, we might expect that similar results would be achieved for any system in which feedback linearization is optimal. Nevertheless, this clearly demonstrates the feasibility as well as computational and performance advantages that may be possible in using MPC+FL. In addition, we would expect that any system where the feedback linearized controller is "close" to the optimal would be particularly amenable to such an MPC+FL algorithm. Complete results are given in Table 2 where the cost column tabulates the value of the original cost.

- **Saturation constraints:** To demonstrate the importance of the explicit consideration of constraints in controller design, the same system was simulated, but with the control saturated at $|u| \leq \alpha$ for

Controller	Cost
OPTIMAL/FL	1.7183
NLMPC ($M = 2,\ P = 5$)	1.8676
MPC+FL ($M = 2,\ P = 5$)	1.8413

Table 2: Example 2; Unconstrained case

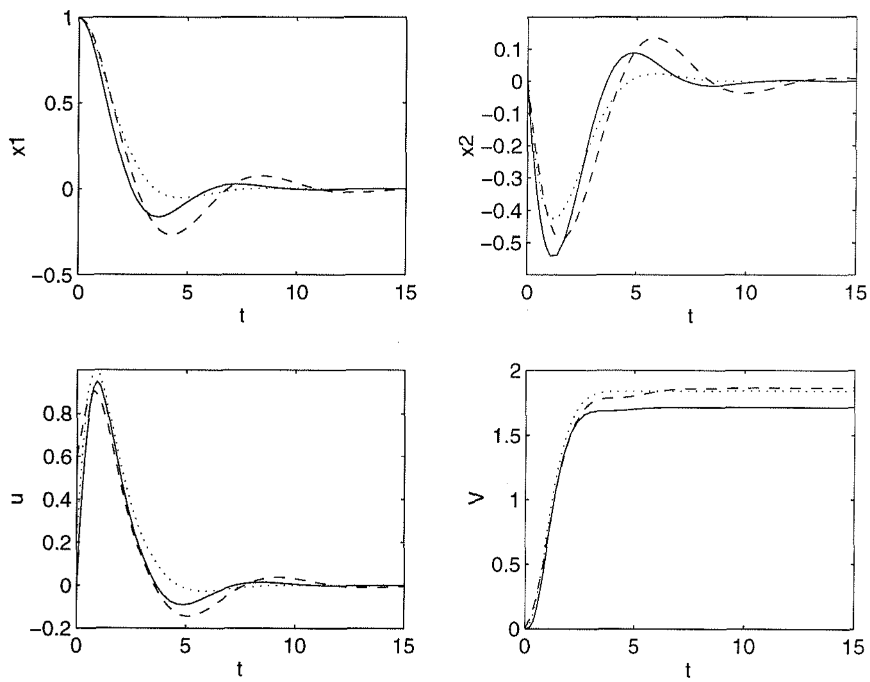

Figure 6: Optimal/FL (solid) vs. NLMPC (dashed) vs. MPC+FL (dotted); Unconstrained case, initial condition [1,0]

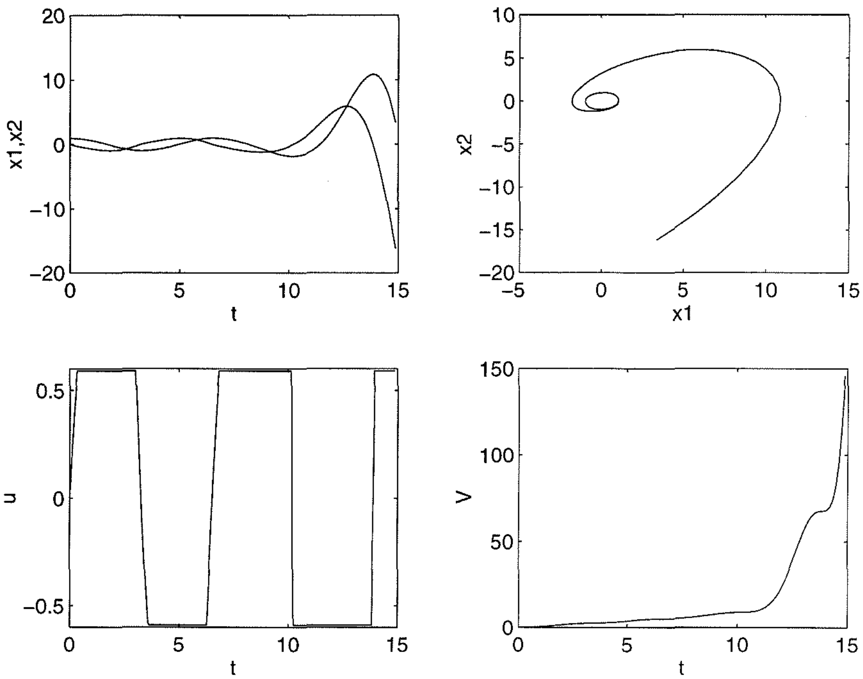

Figure 7: Optimal/FL controller; Saturation $\alpha = 0.59$

differing values of α. The most interesting and instructive turned out to be $\alpha = 0.59$, for which the optimal controller was actually shown to be unstable (Figure 7) with initial condition $[1 \quad 0]$. This instability is due to the fact that the control action increasingly loses effect as the state becomes large (i.e., $\hat{g}(x)$ becomes small as $|x|$ becomes large, see Figure 8). The optimal controller for the unsaturated system does not realize that saturation will limit its ability to control the dynamics if it drifts too far from the origin. NLMPC, when applied to the system with saturation constraints is able to achieve stability, as shown in Figure 9 (solid), but at the high cost of 5.9886. Increasing the horizon lengths from $M = 2,\ P = 5$ to larger values decreases the cost, but at the expense of additional computational burden.

In contrast, the MPC+FL performs extremely well as shown in Figure 9 (dashed). Both the desired cost V and the computational costs are far superior to those in NLMPC, while using the same parameter values of $M = 2,\ P = 5$.

Finally, MPC+FL was applied, but without including the saturation constraint explicitly in the optimization (this is referred to as LTI+FL in Table 3). This resulted in instability, too, despite the "on-line" feedback of MPC type formulations (Figure 10). However, our tests indicate that for some less stringent saturation levels, this method provides satisfactory performance, very close to the constrained MPC algorithm.

This example clearly illustrates the importance of the *explicit* consideration of constraints in control design, and the catastrophic effects that can occur if they are ignored, even when using the optimal controller for the unsaturated system.

In addition, it not only validates the feasibility of the MPC+FL algorithm, but demonstrates its ability to achieve superior performance for certain nonlinear systems even in the presence of constraints, while reducing the associated computational costs.

The results are summarized in Table 3.

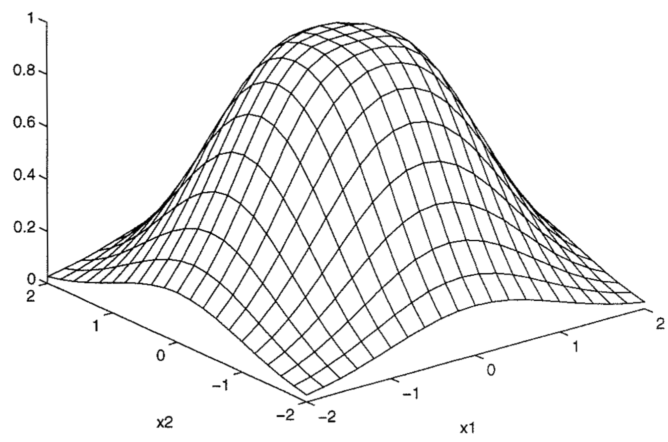

Figure 8: Function $\hat{g}(x)$; Example 2

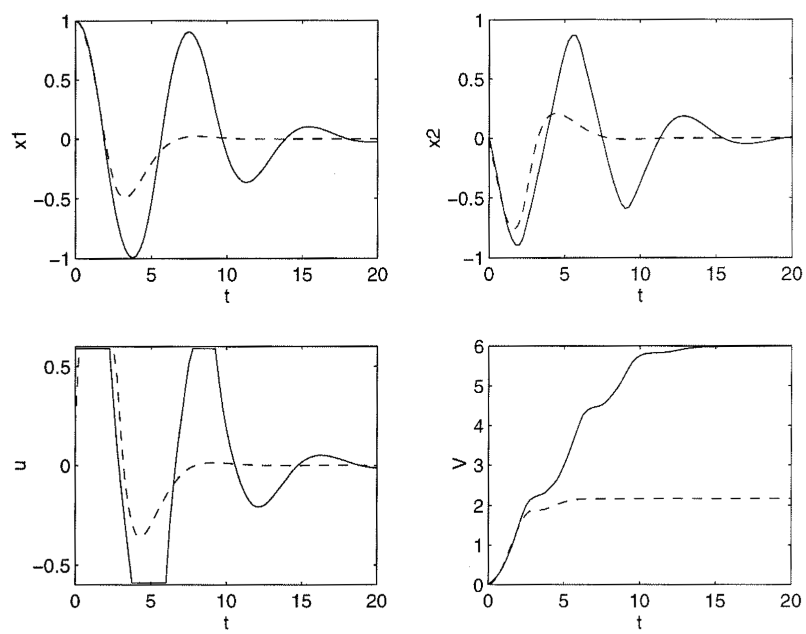

Figure 9: NLMPC (solid) vs. MPC+FL (dashed); Constrained case $\alpha = 0.59$

Controller	Cost
OPTIMAL/FL	unstable
NLMPC ($M = 2,\ P = 5$)	5.9886
MPC+FL ($M = 2,\ P = 5$)	2.1737
LTI+FL ($M = 2,\ P = 5$)	unstable

Table 3: Example 2; Constrained case

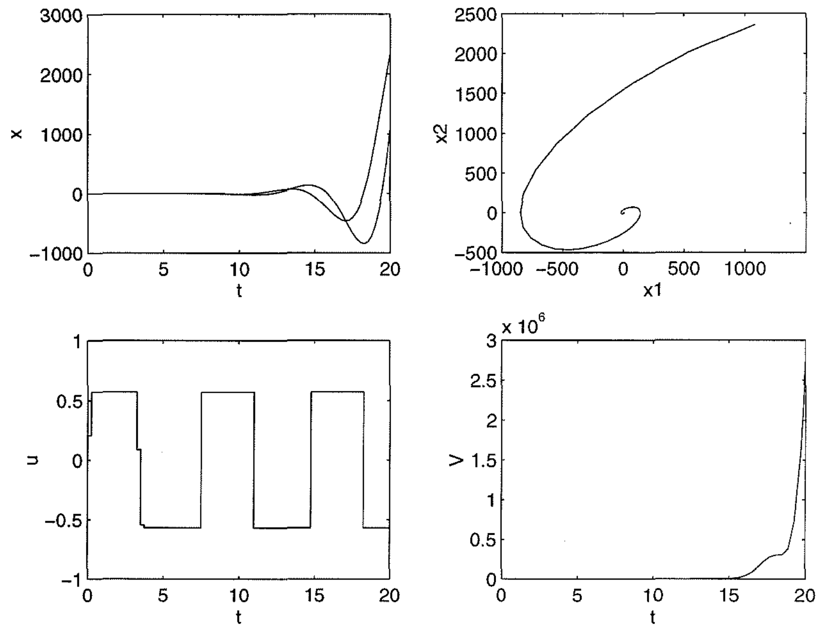

Figure 10: LTI+FL under saturation; $M = 2,\ P = 5,\ \alpha = 0.59$

8.2 Examples: Generated by CoCHJB Approach

In this section various results obtained by applying both NLMPC and MPC+FL to examples generating by solving the converse constrained optimal control problem are presented and evaluated by comparison with the known *constrained optimal controller*, as well as with other control techniques.

8.2.1 Example 1

Consider a system of the form (8) with input saturation constraint $|u| \leq \alpha$. Choosing the value function:

$$V^* = -2(1 + x_2)e^{-x_2} + x_1^2 + 2 \tag{77}$$

we have $V_1^* = 2x_1,\ V_2^* = 2x_2e^{-x_2}$.

Taking $\hat{g}$ to be

$$\hat{g} = e^{2x_1+2x_2}$$

determines $\hat{f}$ from equation (57):

$$\hat{f} = -\frac{(2x_2e^{2x_1+x_2} + u^*)u^* + 2x_1x_2 + x_2^2}{2x_2e^{-x_2}} \tag{78}$$

where u^* is given by:

$$u^* = sat_\alpha(-e^{2x_1+x_2}x_2) \tag{79}$$

The simulation results which follow have been obtained for the initial condition $x_0 = [-5.15\ \ 15]^T$ and saturation $\alpha = 0.5$. For reference, the level curves of $u^* = 0.5$ and $u^* = -0.5$ are given in Figure 11. These curves separate the regions in which the optimal control u^* saturates.

Figure 12 shows a comparison between the constrained optimal nonlinear controller (79) and NLMPC, implemented for two different horizon lengths. State responses x_1 and x_2 are given in the top figures, and

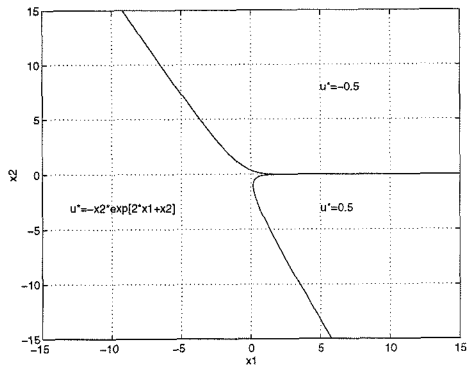

Figure 11: Level curves of $u_o^* = \pm 0.5$; Example 1

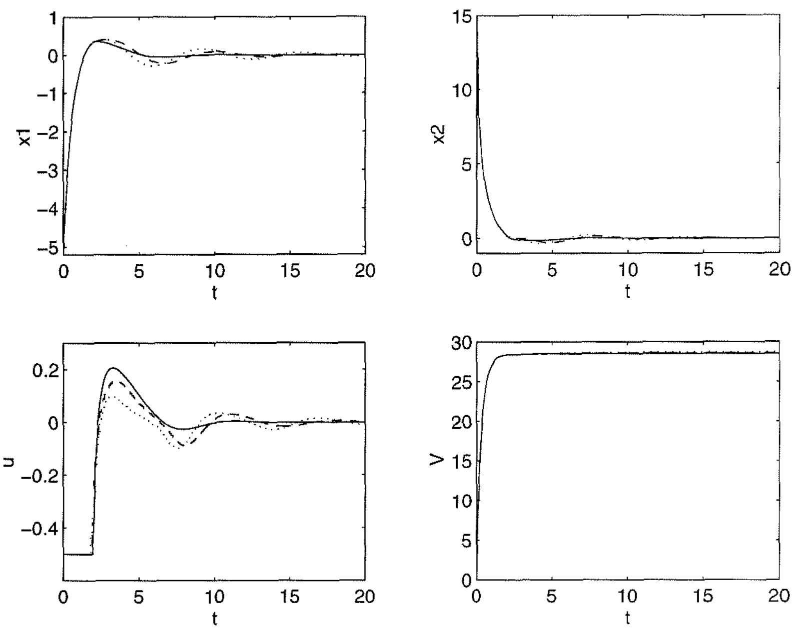

Figure 12: Constrained Optimal Controller (solid) vs. Nonlinear MPC (dashed: M=P=5, dotted: M=P=3)

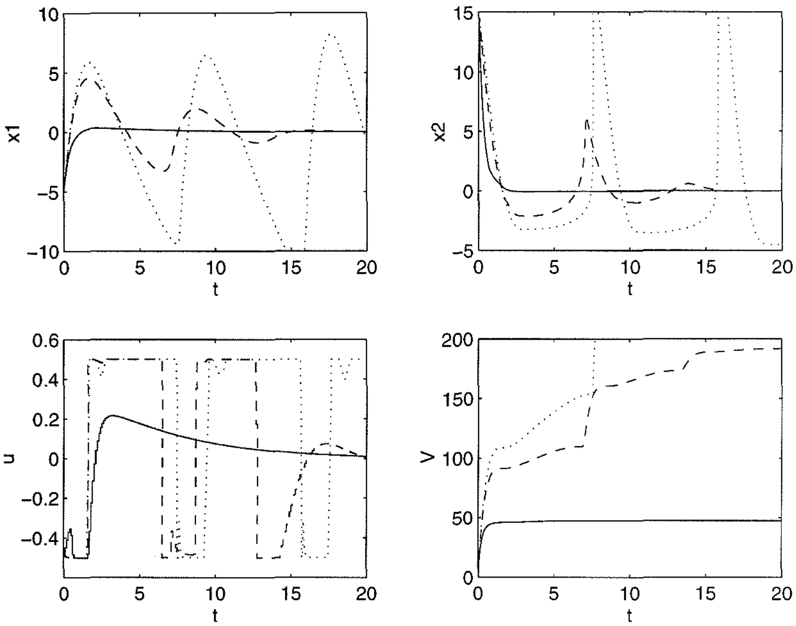

Figure 13: FL (dot) vs. MPC+FL (solid: R=0.1, dashed: R=1)

control action u and cost V in the bottom figures. The performance obtained by NLMPC is nearly identical to that of the constrained optimal controller u^*. This is not surprising since NLMPC includes the constraint directly in its formulation, and deviations from the optimal performance are most likely attributable to the finite horizon lengths or possible failures in the on-line optimization. Also as expected, the larger the MPC horizon used, the nearer the cost is to the optimal (see also Table 4). However, the price for such near optimal performance is the large computational burden associated with NLMPC. Therefore, an important question is whether the expensive computations are justified and how well other less demanding control techniques will fair when applied to this system.

An obvious alternative choice is a controller designed by feedback linearization (FL). An FL controller designed to have the same closed loop dynamics as the locally optimal LQR solution for the system linearized at the origin is given by: $u_{fl} = -(1/\hat{g})(\hat{f} + x_1 + x_2)$. Note that this design does not take the constraint into account. For the same initial condition and saturation level as was used previously, the FL controller applied to the system with saturation results in instability, as shown in Fig. 13 (dotted line). The unstable performance is due to the fact that linearity in the FL controlled closed loop is not preserved during saturation. More specifically, the FL controller fails to take advantage of its full control potential at the time the system is sensitive to the control action, i.e., when $\hat{g}$ is large. Later, saturation and small $\hat{g}$ conspire to diminish the attempts of any control action. In contrast to the FL controller, the optimal controller recognizes that saturation will limit its ability to influence the dynamics in certain portions of the state space, and responds accordingly by saturating fully when in the appropriate regions. In particular, this can be seen at the initial condition where $\hat{g} = e^{19.7}$, and the difference between $u_{fl} = -0.4317$ and $u^* = -0.5$ has a tremendous effect on the closed loop dynamics ($\hat{f} + \hat{g}u_{fl} = -9.85$ vs. $\hat{f} + \hat{g}u^* = -2.45 \times 10^7$).

In order to avoid the difficulties due to saturation, MPC is applied to the FL system according to the MPC+FL formulation. Since the constraint is now explicitly respected, the FL system is protected from instability. Despite the fact that the objective (61) minimized in MPC+FL is not the same as the original objective (58) used in NLMPC, satisfactory performance is achieved (Figure 13). To evaluate the MPC+FL results, the cost measured by the original objective (58) is used, as shown in Figure 13 and given in Table 4. Simulation results clearly indicate that the choice of control weight R in the MPC+FL objective (61) can

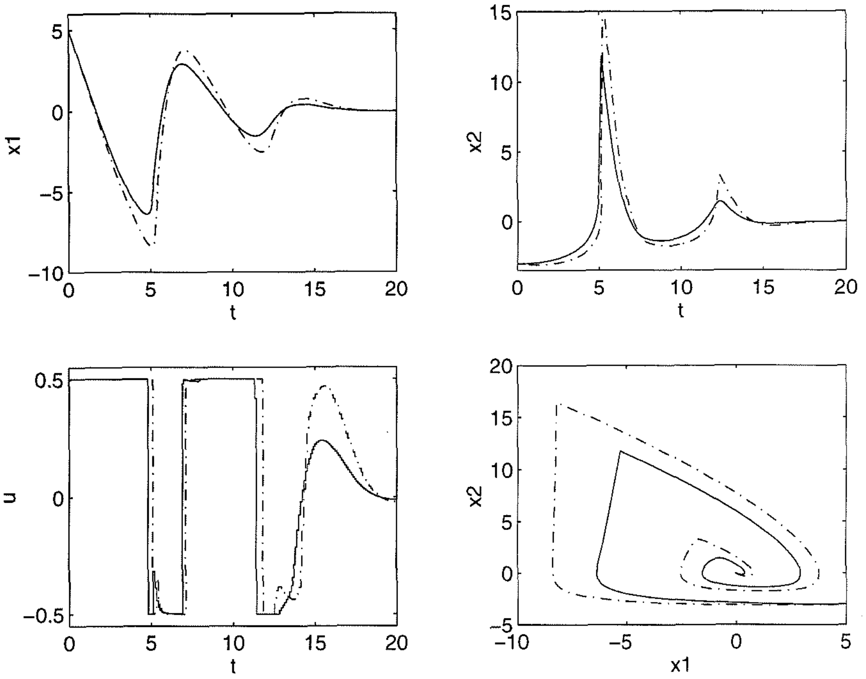

Figure 14: MPC+FL (solid) vs. LTI+FL (dashed): $M = 2,\ P = 10,\ \alpha = 0.5$

be crucial, i.e., careful tuning and an appropriate choice of R can lead to performance close to the optimal. In this example, decreasing the weight R improves the performance.

Other way of improving performance for MPC+FL could be by increasing the length of the prediction horizon and decreasing at the same time the length of the control horizon. The responses for different ratio P/M are shown in the Figure 15. It is clear that this tuning is less efficient than adjusting the weight R discussed before.

Finally, the effect of explicitly considering the constraints in the MPC+FL formulation was judged by comparing the MPC+FL controller with a controller using an unconstrained MPC+FL formulation, despite the actual presence of the constraints (i.e., if the unconstrained MPC control action is applied to the FL loop with saturation; we refer to this as LTI+FL). Because of the on-line implementation of MPC, compared to other techniques it tends to respond better in many situations to unknown constraints and differences between the plant and model. The response of the system under the LTI+FL controller is shown in Figure 14 (dashed line) for the initial condition $x_0 = [5 - 3]^T$ and MPC parameters $M = 2,\ P = 10$. It results in a stable system, whereas the FL controller which similarly ignored the constraints could not stabilize the system. Nevertheless, the performance of the LTI+FL controller is poor in comparison with that of the MPC+FL controller (Figure 14 solid line). For the given initial condition, the cost of the LTI+FL controller is nearly twice that of the MPC+FL controller. In fact, there exist systems in which disregarding the constraints can lead to instability for an LTI+FL controller, as shown in the previous example.

The results for Example 1 and initial condition $x_0 = [-5.15\ \ 15]^T$ are summarized in Table 4[5].

This example clearly shows the necessity of using MPC-based techniques (NLMPC and MPC+FL) for such types of systems.

[5]For this initial condition and for the same tuning parameters, LTI+FL performs and achieves costs very similar to MPC+FL.

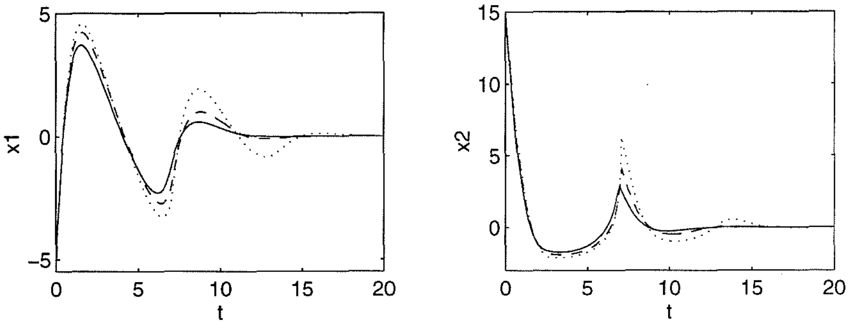

Figure 15: MPC+FL: various output horizons (M=2, solid: P=100, dashed: P=20, dotted: P=5)

Controller	Cost
OPTIMAL	28.5225
NLMPC ($M = P = 5$)	28.6208
NLMPC ($M = P = 3$)	28.7281
MPC+FL ($M = P = 5, R = 0.1$)	47.2107
MPC+FL ($M = P = 5, R = 1$)	191.8618
FL	unstable

Table 4: Example 1; Results

8.2.2 Example 2

In contrast to Example 1, which demonstrates a "worst case" scenario for the FL design that ignores the saturation constraint, in this example it is shown that the opposite extreme is also possible, i.e., saturating a controller designed by feedback linearization may actually make it perform closer to the optimal.

Using $V^* = x_1^2 + x_2^2$ as the value function, and choosing $\hat{g} = e^{x_2}$, according to the procedure proposed in 5.3, the following optimal controller and dynamics are obtained:

$$u^* = sat_\alpha(-x_2 e^{x_2}) \tag{80}$$

$$\hat{f} = -\frac{(2x_2 e^{x_2} + u^*)u^* + 2x_1 x_2 + x_2^2}{2x_2} \tag{81}$$

As will be shown, different from the FL controller in Example 1, in this example the saturation constraint actually protects the FL controller against unnecessarily expensive control action. An FL controller designed for this system that does not undergo saturation uses a large amount of control effort, differing greatly from the optimal controller which uses only a small amount, and results in a cost that exceeds 400. When saturation is applied, it prevents the FL controller from using such an excessive amount of control, and actually reduces the cost to a near optimal value of 25.0098. A plot of the resulting trajectories and control action from the saturated FL controller for a saturation level of $\alpha = 0.2$, and initial condition $[5 \ \ 0]$ is given in Figure 17 (solid line).

It is interesting to note that the major difference between the saturated FL controller and the optimal constrained controller (Figure 16: solid line) is that during the first couple seconds of control actions, the FL controller is always saturated while the constrained optimal control is not. This is in contrast to Example 1 where the constrained optimal control was saturated when the FL controller was not.

MPC techniques using parameter values of $M = 2$ and $P = 20$ were also applied to this example. As shown in Figure 16 where the optimal and NLMPC controller are simultaneously plotted, NLMPC also performs close to optimal, achieving a cost of 25.17 as opposed to the optimal value of 25. Surprisingly, this

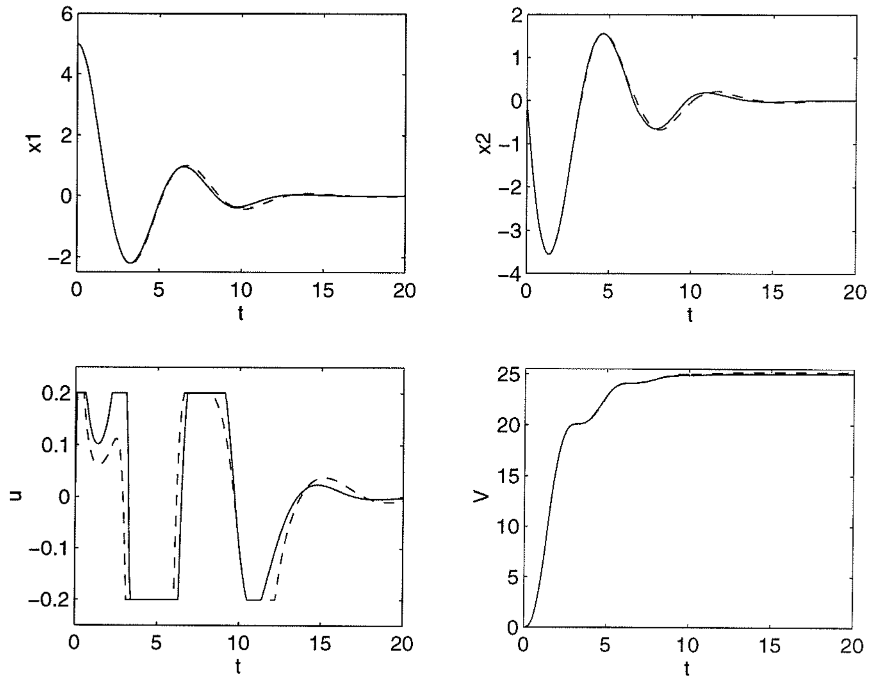

Figure 16: Optimal controller (solid line) vs. NLMPC (dashed); $\alpha = 0.2$

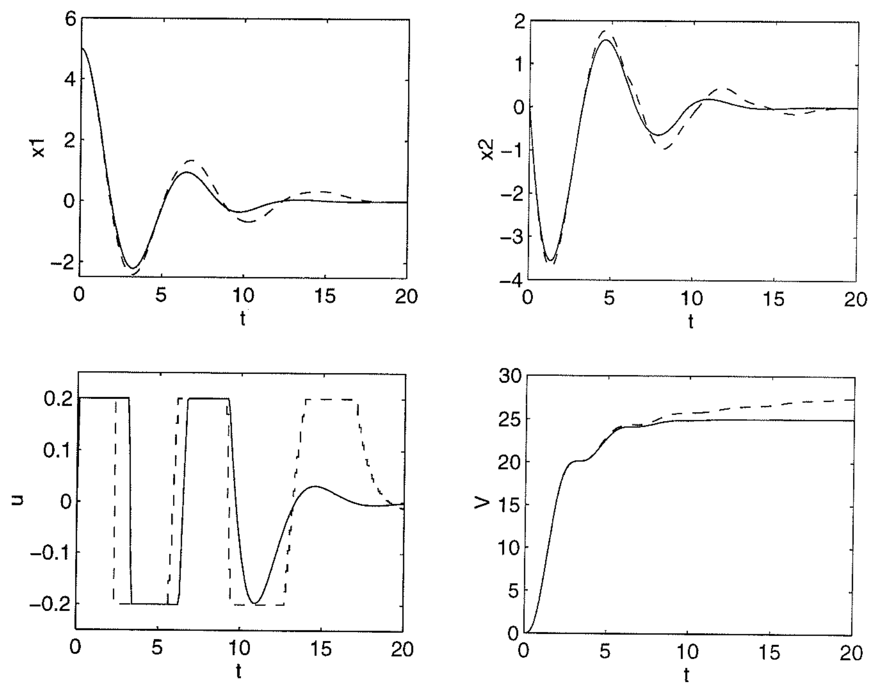

Figure 17: Saturated FL controller (solid) vs. MPC+FL (dashed); $\alpha = 0.2$

Controller	Cost
OPTIMAL	25.00
NLMPC	25.17
FL	25.0098
MPC+FL	27.4182

Table 5: Example 2; Results

is still larger than the cost achieved by the saturated FL controller. Inspection of Figure 16 shows that the shape of the NLMPC control action is qualitatively much closer to that of the optimal than is the control action of the saturated FL controller (Figure 17). This is especially apparent over the first couple seconds of control action.

While the computational effort required of NLMPC is rather large for problems of this sort, the MPC+FL formulation can be used to reduce this burden. A direct comparison of the MPC+FL results with those of the saturated FL controller are shown in Figure 17. Owing to the underlying structure of MPC+FL, which uses the feedback linearized prediction model, we find that the MPC+FL control action is qualitatively quite similar to that of the saturated FL control. MPC+FL results in a cost of 27.4182. While this cost is reasonable when compared to the optimal cost, it is clearly inferior to both NLMPC and saturated FL. This suboptimality can be attributed to the use of the modified cost which is optimized in MPC+FL. Nevertheless, the performance is certainly acceptable and the computational expenditure is reasonable.

Table 5 summarizes the results from Example 2 obtained for $\alpha = 0.2$ and initial condition $x_0 = [5 \quad 0]^T$, with $M = 2,\ P = 20$ in the MPC algorithms:

Based on the discussion above, we can conclude that there exist systems, such as the one explored here, where applying computationally expensive MPC algorithms is unnecessary, even in the presence of constraints.

8.2.3 Brief discussion

Examples 1 and 2 presented in section 8.2 demonstrate the unpredictability of ignoring constraints in nonlinear control design. While in the second example, fortuitously the control saturation constraint works to the advantage of the FL control design, Example 1 shows that a similar disregard for saturation can be disastrous. These two examples are extreme by any measure, but the point is that any range of phenomena in between can occur. We are limited by our system generation technique, the converse constrained HJB approach, which allows us to know both a system with constraints, and the optimal control for that system. Nevertheless, it would be wishful thinking to assume that similar phenomena to those shown above cannot occur in more "reasonable" and "physically motivated" systems.

We were able to show that the design techniques based on MPC, which can handle saturation constraints, provided consistent, near-optimal controllers for both of these systems, despite the high levels of nonlinearity demonstrated. At the expense of computational burden, reliable control from MPC schemes can be achieved. In addition, the choice between NLMPC and MPC+FL represents, in some sense, a trade-off between optimality and computational burden. While not uniformly valid, this trade-off is applicable as a general rule of thumb. Hence, MPC demonstrated itself as a flexible design tool for nonlinear systems that is not only capable of dealing with otherwise intractable constraints, but also allows the control designer a certain variation in the optimality vs. computational trade-off that may be used to fit the needs at hand.

9 Conclusion

Various examples for which the (un)constrained optimal controller is known were generated using the (un)constrained converse strategy, and used to test and validate the performance of different control techniques in the presence of constraints. MPC, which explicitly handles constraints, worked well on these examples with cost close to optimal. Practical implementation of MPC is dominated by the computational burden of both finding the control action on-line and evaluating the scheme off-line by simulation. For a class of FL nonlinear systems, MPC+FL strategy appears to be attractive due to its relative computational efficiency and represents a viable alternative approach to the traditional NLMPC strategy.

The examples described here are somewhat extreme and not necessarily representative of practical applications, yet they fulfill the aim of creating low order nonlinear problems that illustrate specific phenomena, and help to deepen our understanding of nonlinear control. Although the examples in this report do not have explicit physical motivation, it is not hard to generate examples which have similar characteristics and can be given physical motivation.

References

[1] A. E. Bryson and Y. Ho. *Applied Optimal Control.* Hemisphere Publ. Corp., Washington D.C., 1975.

[2] J. C. Doyle, J. Primbs, B. Shapiro, and V. Nevistić. Nonlinear games: examples and counterexamples. In *Proceedings of the* 35^{th} *IEEE Conference on Decision and Control* , pages 3915–3920, Kobe, Japan, December 1996.

[3] V. Nevistić and M. Morari. Constrained control of feedback-linearizable systems. In *Proceedings of the European Control Conference*, pages 1726–1731, Rome, Italy, 1995.

[4] V. Nevistić and J. Primbs. Model predictive control: breaking through constraints. In *Proceedings of the* 35^{th} *IEEE Conference on Decision and Control* , pages 3932–3937, Kobe, Japan, December 1996.

[5] J. Primbs and V. Nevistić. Optimality of nonlinear design techniques: A converse HJB approach. Technical Report No. CIT-CDS 96-022, California Institute of Technology, Pasadena, CA 91125, 1996.